

IMPLEMENTASI *DEEP LEARNING* BERBASIS TENSORFLOW UNTUK PENGENALAN SIDIK JARI



**Disusun sebagai salah satu syarat menyelesaikan Program Studi Strata I pada Jurusan
Informatika Fakultas Komunikasi dan Informatika**

Oleh:

ROYANI DARMA NURFITA

L 200 140 038

**PROGRAM STUDI INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
2018**

HALAMAN PERSETUJUAN

**IMPLEMENTASI *DEEP LEARNING* BERBASIS TENSORFLOW UNTUK
PENGENALAN SIDIK JARI**

PUBLIKASI ILMIAH

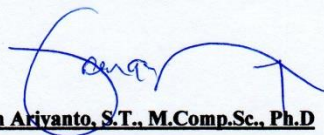
oleh:

ROYANI DARMA NURFITA

L 200 140 038

Telah diperiksa dan disetujui untuk diuji oleh:

Dosen Pembimbing



Gunawan Ariyanto, S.T., M.Comp.Sc., Ph.D

NIK. 968

HALAMAN PENGESAHAN

**IMPLEMENTASI *DEEP LEARNING* BERBASIS TENSORFLOW UNTUK
PENGENALAN SIDIK JARI**

OLEH

ROYANI DARMA NURFITA

L 200 140 038

Telah dipertahankan di depan Dewan Penguji
Fakultas Komunikasi dan Informatika
Universitas Muhammadiyah Surakarta
Pada hari Sabtu, 26 Mei 2018
dan dinyatakan telah memenuhi syarat

Dewan Penguji:

1. Gunawan Ariyanto, S.T., M.Comp.Sc., Ph.D.
(Ketua Dewan Penguji)
2. Heru Supriyono, S.T., M.Sc., Ph.D.
(Anggota I Dewan Penguji)
3. Nurgiyatna, S.T., M.Sc., Ph.D
(Anggota II Dewan Penguji)

(.....)
(.....)
(.....)

**Dekan
Fakultas Komunikasi dan Informatika**



Nurgiyatna, S.T., M.Sc., Ph.D.

NIK. 881

**Ketua Program Studi
Informatika**



Heru Supriyono, S.T., M.Sc., Ph.D.

NIK. 970

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam naskah dan disebutkan dalam daftar pustaka.

Apabila kelak terbukti ada ketidakbenaran dalam pernyataan saya di atas, maka akan saya pertanggungjawabkan sepenuhnya.

Surakarta, 26 Mei 2018

Penulis



ROYANI DARMA NURFITI

L 200 140 038



**UNIVERSITAS MUHAMMADIYAH SURAKARTA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
PROGRAM STUDI INFORMATIKA**

Jl. A Yani Tromol Pos 1 Pabelan Kartasura Telp. (0271)717417, 719483 Fax (0271) 714448
Surakarta 57102 Indonesia. Web: <http://informatika.ums.ac.id>. Email: informatika@ums.ac.id

SURAT KETERANGAN LULUS PLAGIASI

No Surat 227/A.B.-II-3/Inf-FKI/VI/2018

Assalamu'alaikum Wr. Wb

Biro Skripsi Program Studi Informatika menerangkan bahwa :

Nama : Royani Darma Nurfitra
NIM : L200140038
Judul : IMPLEMENTASI *DEEP LEARNING* BERBASIS TENSORFLOW
UNTUK PENGENALAN SIDIK JARI
Program Studi : Informatika
Status : **Lulus**

Adalah benar-benar sudah lulus pengecekan plagiasi dari Naskah Publikasi Skripsi, dengan menggunakan aplikasi Turnitin.

Demikian surat keterangan ini dibuat agar dipergunakan sebagaimana mestinya.

Wassalamu'alaikum Wr. Wb

Surakarta, 9 juni 2018

Biro Skripsi Informatika

Ihsan Cahyo Utomo, S.Kom., M.Kom.



UNIVERSITAS MUHAMMADIYAH SURAKARTA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
PROGRAM STUDI INFORMATIKA

Jl. A Yani Tromol Pos 1 Pabelan Kartasura Telp. (0271)717417, 719483 Fax (0271) 714448
Surakarta 57102 Indonesia. Web: <http://informatika.ums.ac.id>. Email: informatika@ums.ac.id

Secure | https://ev.turnitin.com/app/carta/en_us/?u=1057556090&lang=en_us&o=973920336&cs=1

feedback studio

IMPLEMENTASI DEEP LEARNING BERBASIS TENSORFLOW UNTUK PENGENALAN SIDIK JARI

-- /0

< 5 of 16 > ?

Match Overview

11%

1	repository.ums.ac.id	4%
2	media.neliti.com	2%
3	"Computational Science..."	1%
4	repository.its.ac.id	1%
5	Submitted to iGroup	1%
6	Submitted to De Montf...	1%
7	Submitted to Universita...	1%

IMPLEMENTASI DEEP LEARNING BERBASIS TENSORFLOW UNTUK PENGENALAN SIDIK JARI

Royani Darna Nurrita, Gunawan Ariyanto

Abstrak

Sistem pengenalan sidik jari banyak digunakan dalam bidang biometrik untuk berbagai keperluan pada beberapa tahun terakhir ini. Pengenalan sidik jari digunakan karena memiliki pola yang rumit yang dapat mengahui seseorang dan merupakan identitas setiap manusia. Sidik jari juga banyak digunakan sebagai verifikasi maupun identifikasi. Pemasalahan yang dihadapi dalam penelitian ini adalah komputer sulit melakukan klasifikasi objek salah satunya pada sidik jari. Dalam penelitian ini penulis menggunakan *deep learning* yang menggunakan metode *Convolutional Neural Network (CNN)* untuk mengatasi permasalahan tersebut. *CNN* digunakan untuk melakukan proses pembelajaran mesin pada komputer. Tahapan pada *CNN* adalah input data, *preprocessing*, proses *training*. Implementasi *CNN* yang digunakan pada penelitian ini adalah *library tensorflow* dengan menggunakan bahasa pemrograman python. Dataset yang digunakan bersumber dari sebuah website kompetisi verifikasi sidik jari pada tahun 2004 yang menggunakan sensor ber tipe optical sensor "V300" by CrossMatch dan didalamnya terdapat 80 gambar sidik jari. Proses pelatihan menggunakan data yang berukuran 24x24 pixel dan melakukan pengujian dengan menbandingkan jumlah epoch dan *learning rate* sehingga diketahui bahwa jika semakin besar jumlah epoch dan semakin kecil *learning rate* maka semakin baik tingkat akurasi pelatihan yang didapatkan. Pada penelitian ini tingkat akurasi jelatihan yang dicapai sebesar 100%.

Page: 1 of 13 Word Count: 2410

Text-only Report High Resolution

IMPLEMENTASI *DEEP LEARNING* BERBASIS TENSORFLOW UNTUK PENGENALAN SIDIK JARI

Royani Darma Nurfitra, Gunawan Ariyanto

Abstrak

Sistem pengenalan sidik jari banyak digunakan dalam bidang biometrik untuk berbagai keperluan pada beberapa tahun terakhir ini. Pengenalan sidik jari digunakan karena memiliki pola yang rumit yang dapat mengenali seseorang dan merupakan identitas setiap manusia. Sidik jari juga banyak digunakan sebagai verifikasi maupun identifikasi. Permasalahan yang dihadapi dalam penelitian ini adalah komputer sulit melakukan klasifikasi objek salah satunya pada sidik jari. Dalam penelitian ini penulis menggunakan *deep learning* yang menggunakan metode *Convolutional Neural Network (CNN)* untuk mengatasi permasalahan tersebut. *CNN* digunakan untuk melakukan proses pembelajaran mesin pada komputer. Tahapan pada *CNN* adalah input data, *preprocessing*, proses *training*. Implementasi *CNN* yang digunakan pada penelitian ini adalah *library* tensorflow dengan menggunakan bahasa pemrograman python. Dataset yang digunakan bersumber dari sebuah website kompetisi verifikasi sidik jari pada tahun 2004 yang menggunakan sensor bertipe optical sensor “V300” by CrossMatch dan didalamnya terdapat 80 gambar sidik jari. Proses pelatihan menggunakan data yang berukuran 24x24 pixel dan melakukan pengujian dengan membandingkan jumlah epoch dan *learning rate* sehingga diketahui bahwa jika semakin besar jumlah epoch dan semakin kecil *learning rate* maka semakin baik tingkat akurasi pelatihan yang didapatkan. Pada penelitian ini tingkat akurasi pelatihan yang dicapai sebesar 100%.

Kata Kunci: Pengenalan sidik jari, Deep Learning, Convolutional Neural Network, Tensorflow, Python

Abstract

The fingerprint recognition system is widely used in biometrics for various purposes in recent years. Fingerprint recognition is used because it has a complex pattern that can recognize a person and is the identity of every human being. Fingerprints are also widely used as verification and identification. Problems encountered in this research is the difficult to classify objects one of them on fingerprints. In this study the authors use deep learning using the method of Convolutional Neural Network (CNN) to overcome these problems. CNN is used to perform machine learning process on computer. Stages on CNN are data input, preprocessing, training process. The implementation of CNN used in this research is tensorflow library by using python programming language. The dataset used originated from a fingerprint verification competition website in 2004 using optical sensor type “V300” by CrossMatch and in it there were 80 fingerprint images. The training process uses 24x24 pixel data and performs the test by comparing the number of epoch and learning rate so it is known that if the greater the number of epoch and smaller the learning rate the better the accuracy of the training obtained. In this research, the accuracy level of training is 100%.

Keywords: Fingerprint recognition, Deep Learning, Convolutional Neural Network, Tensorflow, Python

1. PENDAHULUAN

Beberapa tahun terakhir sistem pengenalan sidik jari banyak digunakan dalam bidang teknologi biometrik untuk berbagai keperluan, misalnya pada smartphone, absensi, identifikasi maupun keamanan. Pengenalan sidik jari pada dasarnya adalah sistem pengenalan pola yang

mengenali seseorang dengan menentukan keaslian sidik jarinya. Pengenalan sidik jari dapat digunakan sebagai sistem verifikasi yaitu mengotentikasi identitas seseorang dengan membandingkan citra sidik jari yang diambil dengan citra sidik jari yang telah disimpan di dalam sistem. Tahapan yang dilakukan untuk pengklasifikasian ada dua tahap. Pertama, citra sidik jari yang diinput akan melalui *preprocessing* yang rumit. Kedua, klasifikasi citra sidik jari (Jeon, & Rhee, 2017). Seperti yang dikemukakan oleh Maltoni et al (2009) bahwa klasifikasi sidik jari adalah pengenalan pola yang telah mendapat perhatian besar untuk kesulitannya, karena variabilitas antar-kelas yang kecil dan variabilitas intra-kelas yang besar (Mishelsanti, D., et al, 2017).

Klasifikasi objek pada gambar secara umum menjadi salah satu masalah dalam visi komputer, bagaimana sebuah komputer dapat mencontoh kemampuan manusia dalam memahami informasi gambar, mengenali objek layaknya manusia, seperti mengenali kucing, anjing, pohon, mobil atau mengenali pola tertentu pada sebuah gambar. Bagi manusia hal ini menjadi pekerjaan yang sangatlah sederhana dan mudah, tetapi pada kenyataannya menjadi pekerjaan yang sukar bagi komputer, karena komputer hanya melihat nilai piksel dan data piksel sehingga sulit untuk diproses. Apalagi dengan berbagai variasi dari gambar sangat mempengaruhi pelatihan sehingga untuk mendapatkan hasil yang baik menjadi lebih sulit dan mempengaruhi akurasi. Dengan demikian diharapkan komputer dapat melakukan pengenalan objek layaknya otak biologis manusia walaupun dengan bentuk dan cara kerja yang berbeda. Ketika otak biologis menggunakan jaringan neuron yang memproses perintah, jaringan neuron buatan dirancang berlapis untuk melakukan klasifikasi perintah. Namun, tentu saja otak biologis manusia masih jauh lebih unggul dan kompleks. Maka sistem pengenalan sidik jari pada sebuah gambar ini mengimplementasikan *Deep Learning*.

Deep Learning merupakan bagian dari *Machine Learning* yang terdiri dari banyak lapisan (*hidden layer*) dan membentuk tumpukan, lapisan tersebut adalah sebuah algoritma atau metode yang melakukan klasifikasi perintah yang diinput hingga menghasilkan output. Metode *Deep Learning* yang sedang berkembang salah satunya adalah *Convolutional Neural Network*. Jaringan ini menggunakan masukan berupa gambar, kemudian akan melalui lapisan konvolusi dan diolah berdasarkan filter yang ditentukan, setiap lapisan ini menghasilkan pola dari beberapa bagian citra yang memudahkan proses klasifikasi (Danukusumo, 2017).

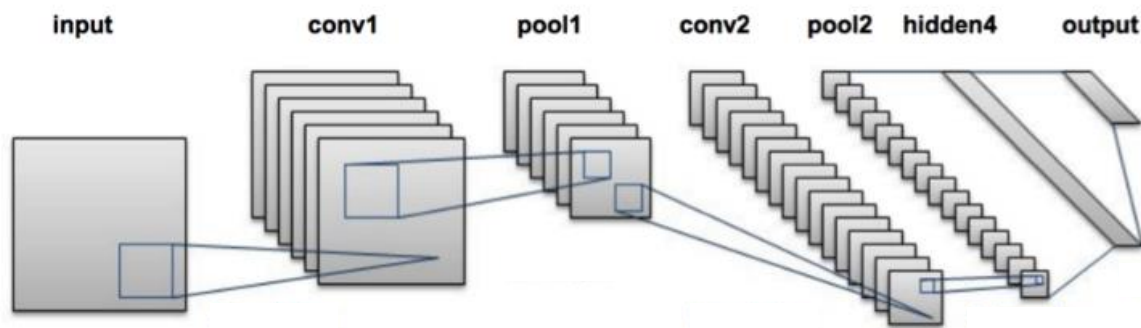
Perkembangan bidang *Deep Learning* saat ini telah dipermudah oleh banyaknya *library* dan *Application Program Interface* (API). *Library* yang digunakan untuk implementasi ini adalah Tensorflow dan menggunakan bahasa pemrograman Python. Tensorflow merupakan antarmuka untuk mengekspresikan algoritma pembelajaran mesin dan untuk mengeksekusi perintah dengan menggunakan informasi yang dimiliki tentang objek tersebut atau target yang dikenali serta dapat

membedakan objek satu dengan objek lainnya. Tentu saja diperlukan data untuk mengenal sebuah gambar yang banyak dan kompleks. Sistem ini sangatlah tepat, fleksibel dan dapat digunakan untuk mengekspresikan berbagai macam algoritma, termasuk algoritma pelatihan dan inferensi untuk model jaringan syaraf. Tensorflow memiliki fitur untuk menjalankan pelatihan model menggunakan *Central Processing Unit* (CPU) dan pelatihan model *Graphic Processing Unit* (GPU) yang memiliki waktu pelatihan yang lebih cepat dibanding pelatihan model CPU. Namun dalam implementasi ini menggunakan CPU.

2. METODE

2.1 *Convolutional Neural Network*

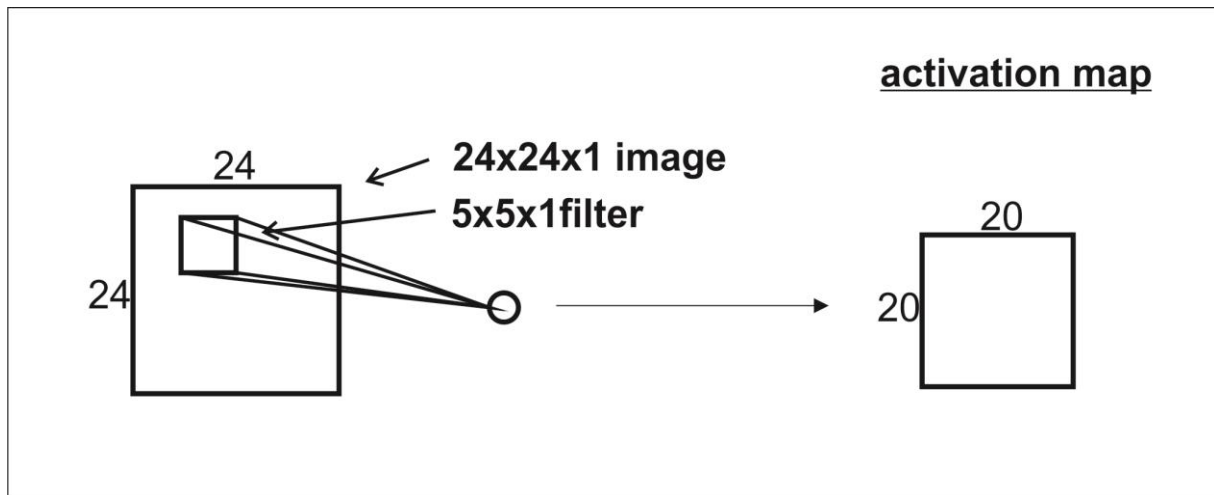
Convolutional Neural Network (CNN) merupakan salah satu pengembangan dari jaringan syaraf tiruan yang terinspirasi dari jaringan syaraf manusia dan biasa digunakan pada data gambar untuk mendeteksi dan mengenali suatu objek pada sebuah gambar. CNN terdiri dari neuron yang memiliki bobot, bias dan fungsi aktivasi. Alur proses CNN seperti Gambar 1.



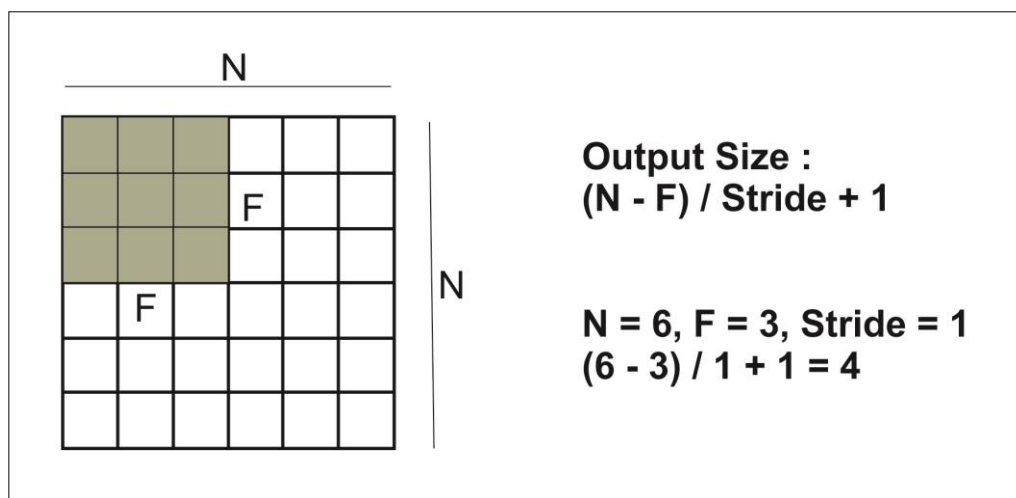
Gambar 1. Proses *Convolutional Neural Network* (Sumber: www.analyticsvidhya.com)

2.1.1 *Convolutional Layer*

Convolutional Layer bagian yang melakukan operasi konvolusi yaitu mengkombinasikan linier *filter* terhadap daerah lokal. *Layer* ini yang pertama kali menerima gambar yang diinputkan pada arsitektur. Bentuk *layer* ini adalah sebuah *filter* dengan panjang (pixel), tinggi (pixel), dan tebal sesuai dengan *channel image* data yang diinputkan. Ketiga *filter* ini akan bergeser keseluruhan bagian gambar. Pergeseran tersebut akan melakukan operasi “dot” antara input dan nilai dari *filter* tersebut sehingga akan menghasilkan output yang disebut sebagai *activation map* atau *feature map*. Gambar 2 menampilkan proses konvolusi yang ada di dalam *convolutional layer* dan Gambar 3 adalah cara menghitung nilai konvolusinya.



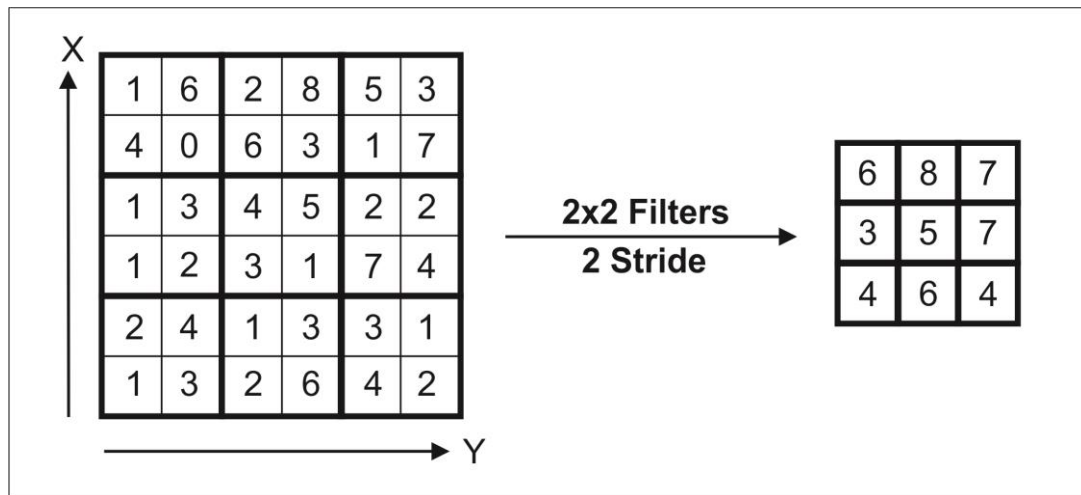
Gambar 2. Proses Konvolusi



Gambar 3. Rumus Menghitung Konvolusi

2.1.2 Pooling Layer

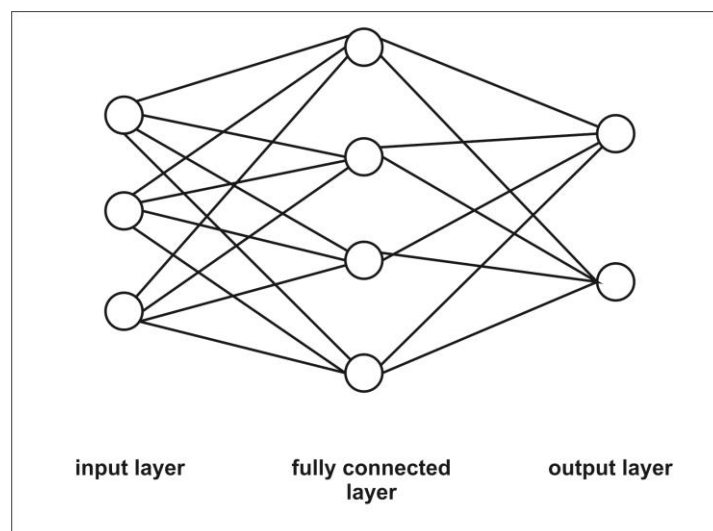
Pooling layer menerima output dari *convolutional layer*, pada *layer* ini ukuran data citra akan direduksi. Prinsipnya *pooling layer* terdiri dari *filter* dengan ukuran tertentu dan *stride/langkah* kemudian bergeser seluruh area *feature map*. Sebagian besar arsitektur CNN, metode *pooling* yang digunakan adalah *max pooling*. *Max pooling* membagi output *convolutional layer* menjadi beberapa grid kemudian setiap pergeseran *filter* akan mengambil nilai terbesar dari setiap grid. Tergantung pada panjang langkahnya, gambar yang dihasilkan adalah sebagian kecil dari ukuran aslinya yang berguna untuk mengurangi dimensi data, sehingga mengurangi jumlah parameter pada langkah selanjutnya (Shukla, & Fricklas, 2018). Gambar 4 menampilkan proses yang ada pada *pooling layer*.



Gambar 4. Proses *Pooling Layer*

2.1.3 Fully Connected Layer

Fully connected layer mengambil input dari hasil output *pooling layer* yang berupa *feature map*. *Feature map* tersebut masih berbentuk multidimensional array maka lapisan ini akan melakukan *reshape feature map* dan menghasilkan vektor sebanyak n-dimensi dimana n adalah jumlah kelas output yang harus dipilih program. Misalnya lapisan terdiri dari 500 neuron, maka akan diterapkan softmax yang mengembalikan daftar probabilitas terbesar untuk masing-masing 10 label kelas sebagai klasifikasi akhir dari jaringan (Dutt Anuj, & Dutt Aashi, 2017). Gambar 5 menampilkan proses yang ada dalam *fully connected layer*.



Gambar 5. Proses *Fully Connected Layer*

2.2 Perancangan Sistem

2.2.1 Menyiapkan Dataset

Dalam melaksanakan penelitian langkah pertama yang dilakukan adalah menyiapkan dataset yang berupa himpunan gambar sidik jari. Dataset digunakan untuk input yang akan diproses pada sistem. Pada penelitian ini akan menggunakan dataset yang bersumber dari <http://bias.csr.unibo.it/fvc2004/databases.asp> yang merupakan sebuah website kompetisi verifikasi sidik jari pada tahun 2004 yang menggunakan sensor bertipe optical sensor “V300” by CrossMatch berukuran 640 x 480 pixel resolusi 500 dpi yang didalamnya ada 80 gambar sidik jari. Karena proses yang akan dijalani memerlukan gambar yang memiliki tinggi dan lebar yang sama maka setiap gambar akan dipotong menjadi 480 x 480 pixel. Gambar 6 menampilkan contoh gambar dari dataset yang digunakan.



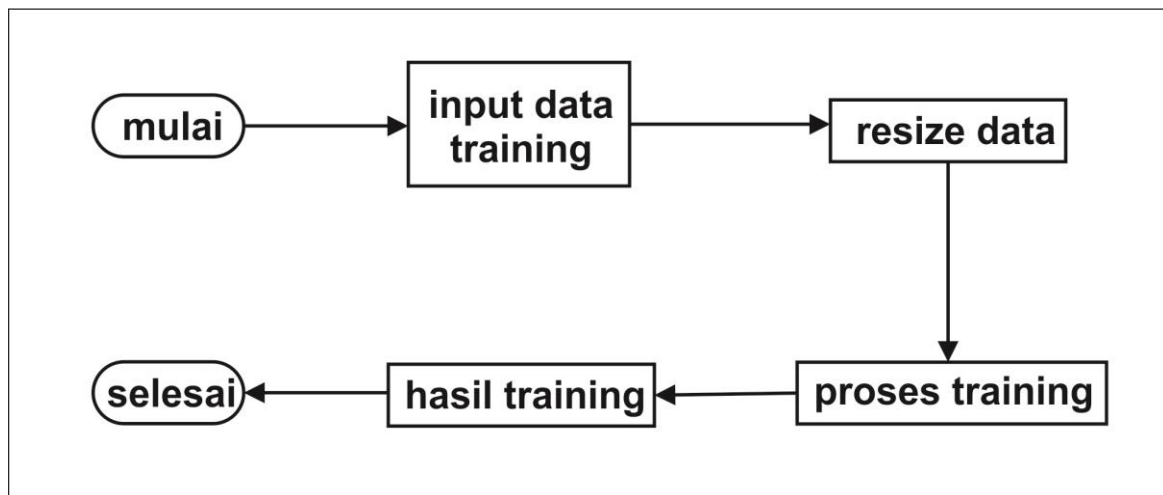
Gambar 6. Contoh dataset sidik jari

2.2.2 Data Preprocessing

Dataset yang digunakan adalah 480 x 480 pixel, ini menjadi ukuran yang sangat besar dan akan menyebabkan sistem terlalu berat untuk melakukan proses pelatihan, maka sebelum memasuki tahap pelatihan dilakukan data *preprocessing* yaitu *resize* gambar menjadi 24x24 pixel.

2.2.3 Pelatihan Dataset

Pada tahap proses ini dataset yang telah siap akan dilatih dengan menggunakan metode *convolutional neural network*. Proses training ini merupakan tahapan dimana CNN dilatih untuk memperoleh akurasi yang tinggi dari klasifikasi yang dilakukan (I Wayan, Wijaya, & Soelaiman, 2016). Alur prosesnya seperti pada Gambar 7.



Gambar 7. Alur Proses *Training*

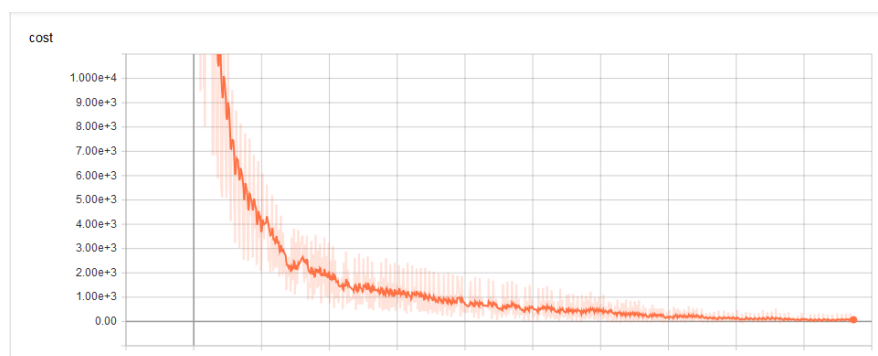
3. HASIL DAN PEMBAHASAN

3.1 Pengujian Jumlah Epoch dan *Learning Rate*

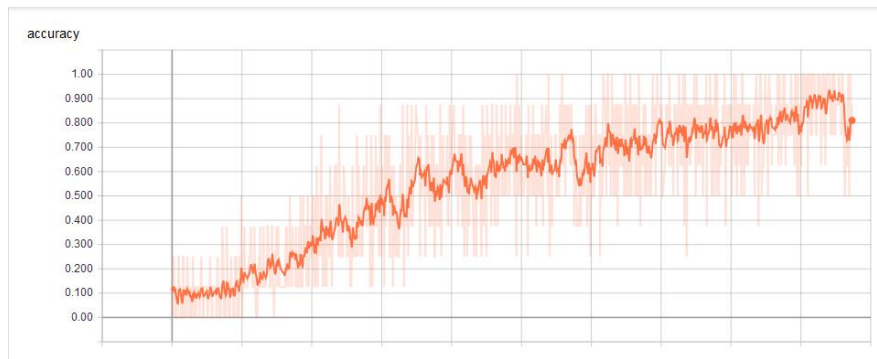
Pengujian dilakukan untuk melihat pengaruh dari jumlah langkah pelatihan (epoch) dan jumlah *learning rate* yang digunakan terhadap performa sistem.

Learning rate sendiri merupakan salah satu parameter *training* untuk menghitung nilai koreksi bobot pada waktu proses *training*, nilai *learning rate* berada pada *range* antara nol (0) sampai satu (1). Semakin besar nilai *learning rate*, maka proses *training* akan berjalan semakin cepat. Namun apabila nilai *learning rate* relatif terlalu besar, maka proses *training* dapat melampaui keadaan optimal yaitu pada saat mencapai nilai *error* yang paling minimal. Dengan kata lain, *learning rate* mempengaruhi ketelitian jaringan suatu sistem (Alfina, 2012).

Sebagai perbandingannya pengujian ini menggunakan jumlah epoch 150 dan epoch 250, sedangkan *learning rate* 0,0001, 0,001 dan 0,01. Hasil pengujian dapat dilihat pada Gambar 8.

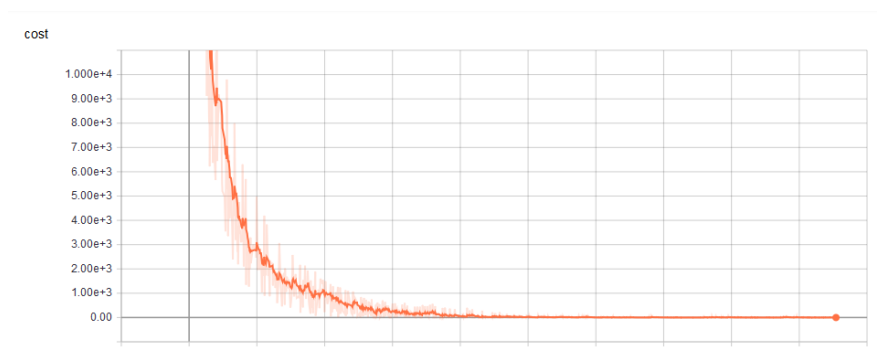


Gambar 8a. Training Loss sistem Epoch 150 dan *Learning rate* 0,0001

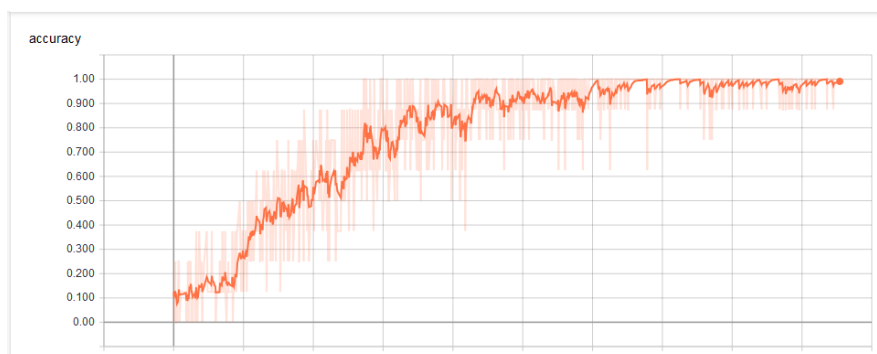


Gambar 8b. Training Accuracy sistem Epoch 150 dan Learning rate 0,0001

Berdasarkan Gambar 8a dan 8b, dapat diketahui bahwa dengan menggunakan langkah pelatihan 150 epoch dan *learning rate* 0,0001 menghasilkan tingkat akurasi 87,50%.



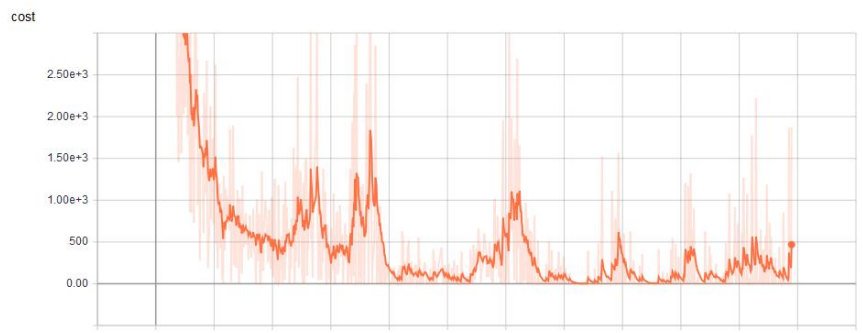
Gambar 9a. Training Loss sistem Epoch 250 dan Learning rate 0,0001



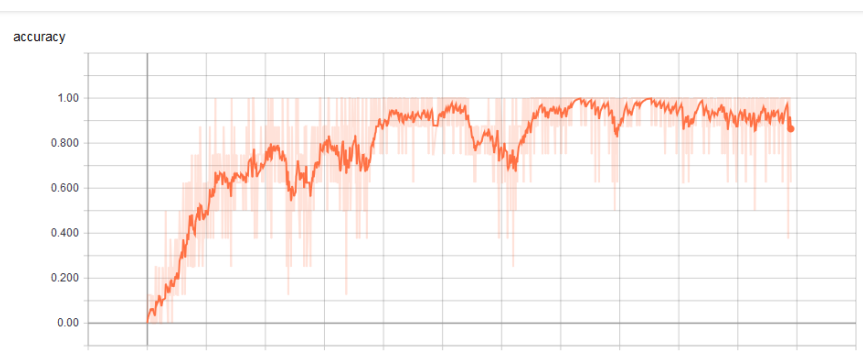
Gambar 9b. Training Accuracy sistem Epoch 250 dan Learning rate 0,0001

Berdasarkan Gambar 9a dan 9b, dapat diketahui bahwa dengan menggunakan langkah pelatihan 250 epoch dan *learning rate* 0,0001 mencapai tingkat akurasi 100%.

Dilihat dari gambar 8 dan 9, proses training yang dilakukan dapat dianalisis bahwa semakin banyak jumlah epoch maka semakin baik persentase akurasi data yang dihasilkan. Namun, semakin panjang jumlah epochnya maka semakin banyak juga waktu yang dibutuhkan untuk proses trainingnya.

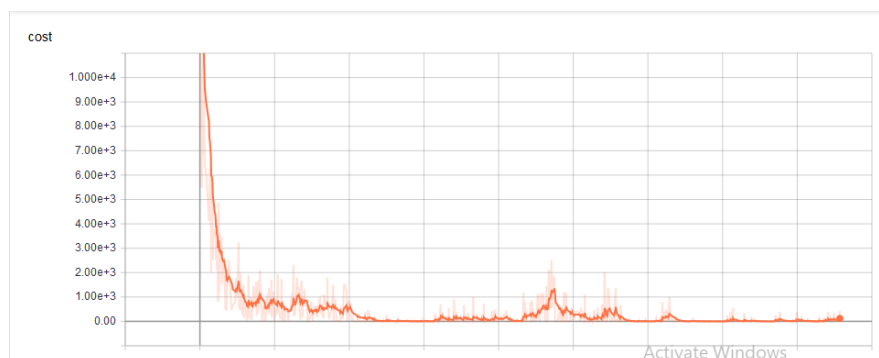


Gambar 10a. Training Loss sistem Epoch 150 dan Learning rate 0,001

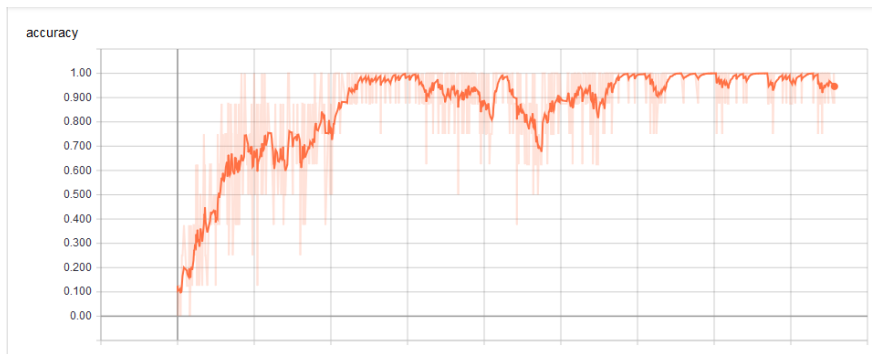


Gambar 10b. Training Accuracy sistem Epoch 150 dan Learning rate 0,001

Hasil training dengan epoch 150 dan *learning rate* 0,001 seperti yang ditampilkan pada Gambar 10a dan 10b mencapai akurasi 62,50%.

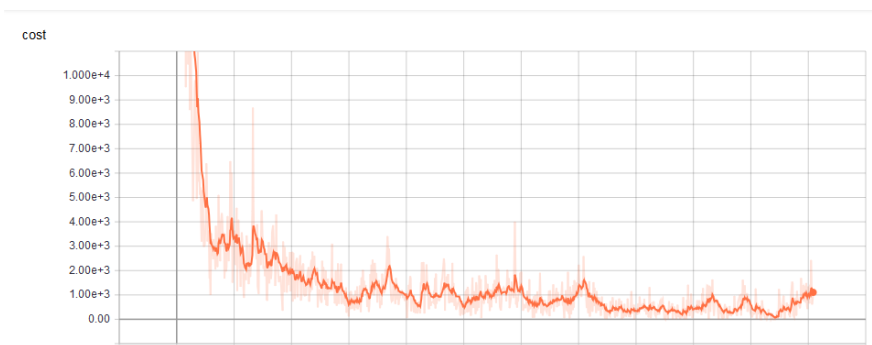


Gambar 11a. Training Loss sistem Epoch 250 dan Learning rate 0,001

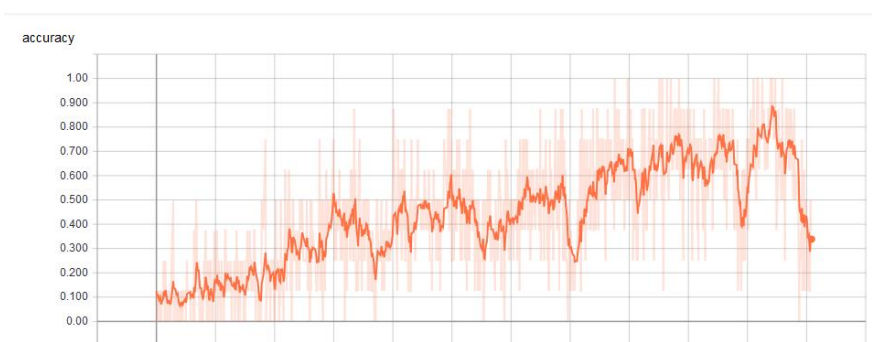


Gambar 11b. Training Accuracy sistem Epoch 250 dan Learning rate 0,001

Pada Gambar 11a dan 11b, dengan menggunakan langkah pelatihan 250 epoch dan *learning rate* 0,001 mencapai akurasi 87,50%.

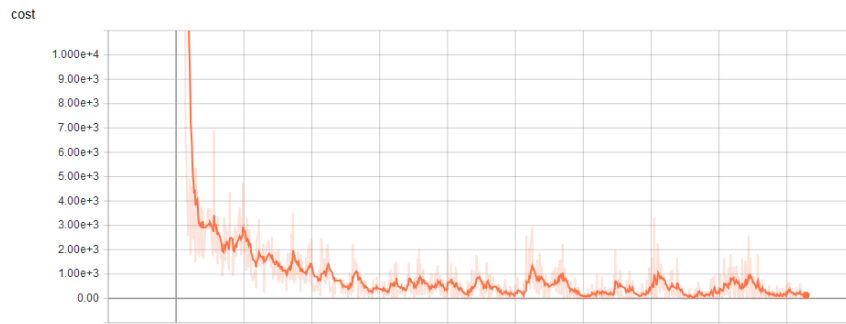


Gambar 12a. Training Loss sistem Epoch 150 dan Learning rate 0,01

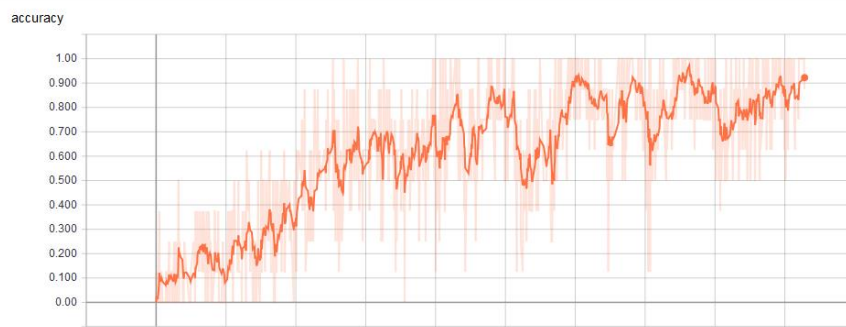


Gambar 12b. Training Accuracy sistem Epoch 150 dan Learning rate 0,01

Berdasarkan Gambar 12a dan 12b, dapat dilihat bahwa dengan menggunakan langkah pelatihan 150 epoch dan *learning rate* 0,01 menghasilkan tingkat akurasi 37,50%.



Gambar 13a. Training Loss sistem Epoch 250 dan Learning rate 0,01



Gambar 13b. Training Accuracy sistem Epoch 250 dan Learning rate 0,01

Berdasarkan Gambar 13a dan 13b, dapat dilihat bahwa dengan langkah pelatihan 250 epoch dan *learning rate* 0,01 menghasilkan tingkat akurasi 87,50%.

Dilihat dari Gambar 12 dan 13, hasil dari persentase proses training yang didapatkan tingkat akurasi dengan epoch yang semakin banyak maka akurasi semakin baik dan dilihat dari Gambar 10, 11, 12 dan 13 bahwa jika *learning rate* semakin besar maka tingkat akurasinya menjadi buruk dan pelatihannya tidak stabil.

3.2 Hasil Pengujian dan Analisis

Pengujian yang dilakukan menggunakan 150 epoch dan 250 epoch dan *learning rate* 0,0001, 0,001 dan 0,01.

Tabel 1. Hasil dan pengujian sistem

Jumlah Dataset	Image Size	Epoch	Learning rate	Accuracy (%)
80	24x24 px	150	0,0001	87,50
		150	0,001	62,50
		150	0,01	37,50

Jumlah Dataset	Image Size	Epoch	Learning rate	Accuracy (%)
80	24x24 px	250	0,0001	100
		250	0,001	87,50
		250	0,01	87,50

Berdasarkan tabel hasil pengujian sistem, dapat diketahui bahwa dataset berjumlah 80 dengan ukuran 24x24 px menggunakan epoch 150 pada saat proses training menggunakan *learning rate* 0,0001 menghasilkan tingkat akurasi 87,50%, menggunakan *learning rate* 0,001 menghasilkan tingkat akurasi 62,50% sedangkan dengan *learning rate* 0,01 menghasilkan tingkat akurasi 37,50%. Dan pengujian dengan epoch 250 menggunakan *learning rate* 0,0001 menghasilkan akurasi 100%, dengan *learning rate* 0,001 menghasilkan akurasi 87,50% dan dengan menggunakan *learning rate* 0,01 juga menghasilkan akurasi 87,50%. Dari perbandingan tersebut dapat diketahui bahwa jumlah epoch dan *learning rate* sangat berpengaruh pada tingkat akurasi. Semakin besar jumlah epoch maka semakin baik tingkat akurasinya, sedangkan jika semakin besar *learning rate* maka semakin buruk hasil akurasinya dan dalam proses pelatihannya tidak stabil. Seperti yang dijelaskan oleh Skapura (1991, h. 104) bahwa semakin besar *learning rate*, maka ketelitian jaringan akan semakin berkurang, dan berlaku sebaliknya. Apabila semakin kecil *learning rate*, maka ketelitian jaringan semakin besar atau bertambah. Namun konsekuensinya adalah proses *training* akan semakin lama (Alfina, 2012).

4.KESIMPULAN

Penelitian ini berhasil mengimplementasikan *Deep Learning* dengan metode *Convolutional Neural Network* dengan *library* tensorflow menghasilkan tingkat akurasi pelatihan yang baik dengan akurasi training sebesar 100%. Proses training yang dilakukan menggunakan beberapa perbandingan antaranya jumlah epoch dan jumlah *learning rate*. Sehingga dapat disimpulkan bahwa jika semakin banyak epoch dan semakin kecil *learning rate* maka semakin baik hasil akurasi pelatihan yang didapatkan.

DAFTAR PUSTAKA

- Alfina, O. (2012). Analisis Perbandingan *Neural Network Backpropagation* Dengan *Simple Perceptron* Dalam Mengenali *Image* Daun. Tesis. Universitas Sumatera Utara.
- Danukusumo, K. P. (2017). Implementasi Deep Learning Menggunakan Convolutional Neural Network Untuk Klasifikasi Citra Candi Berbasis GPU. Skripsi. Universitas Atma Jaya Yogyakarta.
- Dutt, Anuj, & Dutt, Aashi. (2017). Handwritten digit recognition using deep learning. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 6(7), 1-8.
- I Wayan, S. E. P., Wijaya, A. Y., & Soelaiman, R. (2016). Klasifikasi citra menggunakan *Convolutional Neural Network* (CNN) pada Caltech 101. *Jurnal Teknik ITS*, 5(1), 1-5.
- Jeon, W. S., & Rhee, S. Y. (2017). Fingerprint pattern classification using convolutional neural network. *International Journal of Fuzzy Logic and Intelligent Systems*. 17(3), 170-176. <http://dx.doi.org/10.5391/IJFIS.2017.17.3.170>.
- Michelsanti, D., Guichi, Y., Ene, A-D., Stef, R., Nasrollahi, K., & Moeslund, T. B. (2017). Fast Fingerprint Classification With Deep Neural Network. In *VISAPP - International Conference on Computer Vision Theory and Applications*, 1-9.
- Shukla, N., & Fricklas, K. (2018). *Machine learning with tensorflow*. New York:Manning Publications.